

Manual Técnico

Projeto N.º 2 - Época Normal

Inteligência Artificial - Escola Superior de Tecnologia de Setúbal

2023/2024

Prof. Joaquim Filipe

Grupo 17

Nuno Martinho, n.º 201901769

João Coelho, n.º 201902001

João Barbosa, n.º 201901785

Índice

- [Introdução](#)
- [Organização do projeto](#)
- [Tipos Abstratos de Dados](#)
- [Algoritmo Implementado](#)
- [Análise dos Resultados e Limitações](#)
- [Análise Estatística](#)

Introdução

Neste manual técnico é abordada a implementação de um programa em **LISP** que tem como objetivo resolver tabuleiros do **Jogo do Cavalo** (versão simplificada do Passeio do Cavalo), cujo objetivo é, através dos movimentos do cavalo, visitar todas as casas de um tabuleiro similar ao de xadrez. Esta versão decorrerá num tabuleiro de 10 linhas e 10 colunas (10x10), em que cada casa possui uma pontuação.

O objetivo deste será permitir que, em partidas de 2 jogadores (*Humano vs Computador* ou *Computador vs Computador*), estes movimentem as suas peças de xadrez (cavalos) em jogadas sequenciais e que acumulem pontos segundo as regras do jogo. A partida termina quando não for possível movimentar mais essas peças, declarando assim como vencedor aquele que tiver a pontuação mais alta no fim da mesma.

Organização do projeto

O projeto foi implementado no **LispWorks Personal Edition 8.0.1 (64bits)**, um ambiente de desenvolvimento integrado (IDE) para a linguagem de programação *Lisp*. Esta edição pessoal do *LispWorks* é uma versão gratuita e limitada do ambiente completo oferecido pela *LispWorks Ltd*.

Este encontra-se organizado em 3 ficheiros *.lisp*:

- ***interact.lisp*** - corre o programa, lê e escreve em ficheiros e trata da interação com o utilizador.

- **jogo.lisp** - implementação dos operadores do jogo.
- **algoritmo.lisp** - implementação do algoritmo Negamax.

Tipos Abstratos de Dados

Os Tipos Abstratos de Dados desempenham um papel crucial neste projeto, fornecendo uma estrutura organizada e modular para representar e manipular dados. É importante referir a captação dos conceitos do problema e depois estrutura-los para o tipo abstrato de dados. Construíram-se quatro tipos fundamentais:

- Tabuleiro
- Resultado
- No
- Solução

Tabuleiro

O tabuleiro consiste numa matriz de dimensões 10 por 10, composta por 10 linhas, cada uma contendo 10 números variando entre 0 e 99. Estes elementos podem representar números, conforme mencionado, ou assumir os valores -1, que simboliza a peça do cavalo branco, e -2, que simboliza a peça do cavalo preto. Além disso, pode aparecer o valor NIL, indicando que a casa correspondente já não está acessível.

```
<Tabuleiro> ::= ( { Linha }* )
<Linha> ::= { <número> }*
```

Resultado

O resultado é constituído por um tabuleiro, os pontos que o jogador que fez a jogada atual ganhou, e o jogador que fez esta jogada. A função *usar-operadores* devolve uma lista com vários resultados ou seja das jogadas possíveis daquele jogador em específico, que depois esta lista será interpretada pelo algoritmo para transformar cada resultado em um no.

```
<Resultado> ::= ( <tabuleiro> <pontoParaAdicionar> <jogador> )
```

No

O conceito de nó, utilizado nos algoritmos de procura, foi elaborado considerando as exigências do problema. Este conceito engloba o estado atual do problema, representado pelo tabuleiro, o nó pai, o valor atribuído ao nó (função de utilidade) e os pontos dos jogadores 1 e 2.

```
<Nó> ::= ( <tabuleiro> <pai> <f> pontos1 pontos2 )
```

Solução

O algoritmo retorna uma solução, composta pelo melhor nó escolhido para jogar, seguido de uma lista que inclui o número de nós analisados, o número de cortes alfa e beta efetuados, e, por último, o tempo despendido até encontrar essa solução.

```
<Solução>::= (<no-jogada> (<nos-analisados> <n-cortes> <tempo-gasto>))
```

Algoritmo Implementado

Optou-se por implementar o algoritmo *NEGAMAX* com cortes alfa-beta, devido à sua sintaxe mais coesa e de fácil compreensão. Além disso, há a possibilidade de, em alguns casos, ser mais eficiente do que implementações do *MINIMAX* com cortes alfa-beta. A implementação deste algoritmo foi realizada de forma recursiva.

Pseudocódigo NEGAMAX com cortes alfa-beta

```
;; argumentos: nó n, profundidade d, cor c
;;; b = ramificação (número de sucessores)
function negamax (n, d,  $\alpha$ ,  $\beta$ , c)
  se d = 0 ou n é terminal
    return c * valor heurístico de n
  sucessores := OrderMoves(GenerateMoves(n))
  bestValue :=  $-\infty$ 
  para cada sucessor nk em sucessores
    bestValue := max (bestValue, -negamax (nk, d-1,  $-\beta$ ,  $-\alpha$ , -c))
     $\alpha$  := max ( $\alpha$ , bestValue)
    se  $\alpha \geq \beta$ 
      break
  return bestValue
```

Em Lisp:

O algoritmo recebe como entrada um nó, um limite de tempo (em milissegundos), a profundidade máxima (que o valor *default* é 50), a peça com que o algoritmo vai jogar, a cor atual do nó (específico do algoritmo *NEGAMAX*, para determinar se é nó min ou max), os valores de alfa (- infinito) e os de beta (+ infinito), o número de nós analisados, o número de cortes feitos e por fim funções necessárias passadas por parâmetros para o algoritmo ser o mais geral possível.

O código é dividido em duas funções principais: *negamax* e *negamax-aux*.

- *negamax* é a função principal que lida com a profundidade máxima, a ordenação dos sucessores e chama *negamax-aux*.
- *negamax-aux* é a função auxiliar que realiza a recursão e executa os cortes alfa-beta.

O algoritmo encerra a busca quando atinge a profundidade máxima, quando o tempo limite é excedido ou quando não há sucessores disponíveis.

Quando uma solução é encontrada o algoritmo não devolve o valor da função de utilidade diferente do pseudocódigo acima mostrado, porém retorna uma estrutura solução com o melhor nó (nó que tem o melhor valor de utilidade).

```
;; ===== ALGORITMOS =====
(defun negamax (no
                tempo-limite
                &optional
                (d-maximo 50) ;; profundidade default e' 50
                (peca-jogador1 -1) ;; peca do jogador 1 (max)
                (cor 1) ;; começar com no max
                (alfa most-negative-fixnum) ;; começar o alfa a -infinito
                (beta most-positive-fixnum) ;; começar o beta a +infinito
                (tempo-inicial (get-universal-time)) ;; tempo que começa o
algoritmo
                (nos-analisados 1) ;; nos analisados
                (numero-cortes 0) ;; numero de cortes
                (fn-expandir-no 'usar-operadores) ;; funcao geral para expandir o
no
                (fn-selecionar-problema 'resultado-tabuleiro) ;; funcao geral para
ir buscar o tabuleiro do resultado
                (fn-selecionar-pontos 'resultado-pontos) ;; funcao geral para ir
buscar os pontos do resultado
                (fn-inverter-sinal-jogo 'inverter-sinal-tabuleiro)) ;; funcao
geral para inverter o tabuleiro (proprio para o negamax)
  "Executa o algoritmo negamax para um no"
  (let* ((lista-expandidos (ordenar-negamax (gerar-sucessores
                                             no
                                             cor
                                             fn-expandir-no
                                             peca-jogador1
                                             fn-selecionar-problema
                                             fn-selecionar-pontos)
                                             cor))
         (tempo-decorrido (obter-tempo-gasto tempo-inicial))) ;; obter quanto
tempo passou
    (cond
      ((or (= d-maximo 0) (= (length lista-expandidos) 0) (>= tempo-decorrido (/
tempo-limite 1000)))) ;; se e' profundidade maxima, se e' no folha, se passou do
tempo limite
        ;;(criar-no-solucao no nos-analisados numero-cortes tempo-inicial))
        (criar-no-solucao (if (= cor 1) no (inverter-sinal-no no fn-inverter-
sinal-jogo)) nos-analisados numero-cortes tempo-inicial)) ;; devolve o no com a
jogada;; verificar se o no era min se sim, inverter o sinal ;;todo: algo de errado
n esta certo aqui
      (T
        (negamax-aux ;;aplicar o auxiliar do negamax para os sucessores
                     no
                     lista-expandidos
                     tempo-limite
                     d-maximo
                     peca-jogador1
```

```

cor
alfa
beta
tempo-inicial
nos-analisados
numero-cortes
fn-inverter-sinal-jogo))))
;; ===== AUXILIARES NEGAMAX =====
(defun negamax-aux (no-pai
                    sucessores
                    tempo-limite
                    d-maximo
                    peca-jogador1
                    cor
                    alfa
                    beta
                    tempo-inicial
                    nos-analisados
                    numero-cortes
                    fn-inverter-sinal-jogo)
  "Negamax Auxiliar"
  (cond
    ((= (length sucessores) 1) ;; so tem 1 no
      (negamax ;; inverter negamax
        (inverter-sinal-no (car sucessores) fn-
          inverter-sinal-jogo)
          tempo-limite
          (1- d-maximo)
          peca-jogador1
          (- cor)
          (- beta)
          (- alfa)
          tempo-inicial
          (1+ nos-analisados)
          numero-cortes))
      (T
        (let* ((solucao (negamax (inverter-sinal-no (car sucessores) fn-inverter-
          sinal-jogo)
            tempo-limite
            (1- d-maximo)
            peca-jogador1
            (- cor)
            (- beta)
            (- alfa)
            tempo-inicial
            (1+ nos-analisados)
            numero-cortes))
          (no-solucao (car solucao)) ;; vai buscar o no de uma solucao
          encontrada
          (melhor-valor (max-no-f no-solucao no-pai)) ;; compara os valores de f
          e guarda o no com o f mais alto
          (novo-alfa (max alfa (no-f melhor-valor))) ;; atualiza o alfa
          (nos-analisados-solucao (solucao-nos-analisados (cadr solucao))) ;;

```

```

nos analisados da solucao
    (numero-cortes-solucao (solucao-numero-cortes (cadr solucao)))) ;;
numero de cortes da solucao
    (if (>= novo-alfa beta) ;;corte
        (criar-no-solucao (if (= cor 1) no-pai (inverter-sinal-no no-pai fn-
inverter-sinal-jogo))
                            nos-analisados-solucao (1+ numero-cortes-solucao)
tempo-inicial)

;;nao tem corte
(negamax-aux no-pai
              (cdr sucessores)
              tempo-limite
              d-maximo
              peca-jogador1
              cor
              novo-alfa
              beta
              tempo-inicial
              nos-analisados-solucao
              numero-cortes-solucao
              fn-inverter-sinal-jogo))))))

```

A Função **max-no-f** recebe dois nós, analisa qual deles possui o maior valor na função de utilidade (f) e retorna esse nó.

```

(defun max-no-f (no1 no2)
  (let ((value-no1 (no-f no1))
        (value-no2 (no-f no2)))
    (if (> value-no1 value-no2) no1 no2)))

```

A função **ordenar-negamax** é uma implementação do algoritmo de ordenação quicksort adaptada para ordenar uma lista de nós. A função recebe uma lista de nós e uma cor (1 para nós MAX, -1 para nós MIN) como parâmetros. Ela utiliza um comparador que varia com base na cor dos nós para determinar se a ordenação deve ser decrescente (MAX) ou crescente (MIN).

A função principal, **ordenar-negamax**, verifica se a lista de nós é vazia. Se for, retorna uma lista vazia. Caso contrário, seleciona um pivô (o primeiro nó da lista) e divide a lista em duas partes: uma contendo nós maiores que o pivô e outra contendo nós menores. Em seguida, aplica recursivamente o algoritmo quicksort a ambas as partes, concatenando as partes ordenadas.

A função **lista-por-condicao** é um auxiliar que recebe uma condição, um pivô e uma lista de nós. Ela compara os valores dos nós com o valor do pivô usando a condição fornecida e, com base nessa comparação, cria uma nova lista com os nós que atendem à condição.

A ordenação dos nós é benéfica para o algoritmo negamax porque reduz a complexidade da busca. Quando os nós estão ordenados, é mais provável que cortes alfa-beta ocorram mais cedo, pois o algoritmo pode identificar rapidamente se um determinado ramo não levará a uma solução melhor. Além disso, uma boa ordenação pode melhorar o desempenho geral do algoritmo, reduzindo o número de nós que precisam ser analisados. Isso é crucial, especialmente em jogos complexos com grandes árvores de jogo, onde a eficiência do algoritmo de busca é fundamental para a tomada de decisões rápidas.

```
(defun ordenar-negamax (lista-nos cor)
  "Ordena uma lista executando o algoritmo quicksort"
  (if (null lista-nos)
      nil
      (let* ((comparador (if (= cor 1) #'> #'<))
              (pivot (car lista-nos))
              (menores (lista-por-condicao (complement comparador) pivot (cdr
lista-nos)))
              (maiores (lista-por-condicao comparador pivot (cdr lista-nos))))
        (append (ordenar-negamax maiores cor) (list pivot) (ordenar-negamax
menores cor)))))

(defun lista-por-condicao (condicao pivot lista-nos)
  "Auxiliar quicksort para valores de acordo com a condição"
  (if (null lista-nos)
      nil
      (let ((valor (funcall 'no-f (car lista-nos))))
        (if (funcall condicao valor (no-f pivot))
            (cons (car lista-nos) (lista-por-condicao condicao pivot (cdr lista-
nos)))
            (lista-por-condicao condicao pivot (cdr lista-nos)))))
```

Esta função, `gerar-sucessores`, recebe um nó atual, uma cor, uma função de expansão de nós (`fn-expandir-no`), a peça do jogador 1 (`peca-jogador1`), e duas funções (`fn-selecionar-problema` e `fn-selecionar-pontos`) para selecionar o problema (tabuleiro) e os pontos do resultado.

A função utiliza a função de expansão de nós para gerar uma lista de resultados possíveis a partir do tabuleiro atual. Em seguida, mapeia sobre essa lista, criando novos nós para cada resultado possível. Os pontos do jogador 1 e 2 são atualizados com base na cor atual e nos pontos do resultado.

O valor de utilidade (função de avaliação) é calculada da seguinte maneira:

```
$$
|pontos jogador1-pontos jogador 2|
$$
```

```
(defun gerar-sucessores (no-atual cor fn-expandir-no peca-jogador1 fn-selecionar-
problema fn-selecionar-pontos)
  "Recebe um no e a funcao de expansao de nos,
  (a funcao passada normalmente vai ser a usar-operadores que ira gerar uma lista
das proximas jogadas)
```

```

depois essa lista de tabuleiros sera convertida para uma lista de nos."

;;(let ((jogador (if (= cor 1) peca-jogador1 peca-jogador2)))

(mapcar #'(lambda (resultado)
  (let* ((pontos1 (if (= cor 1)
    (+ (no-jogador1 no-atual) (funcall fn-
selecionar-pontos resultado))
    (no-jogador1 no-atual)))
    (pontos2 (if (= cor -1)
    (+ (no-jogador2 no-atual) (funcall fn-
selecionar-pontos resultado))
    (no-jogador2 no-atual))))

    (criar-no
      (funcall fn-selecionar-problema resultado)
      no-atual
      (abs (- pontos1 pontos2))
      pontos1
      pontos2)))
  (funcall fn-expandir-no (no-tabuleiro no-atual) peca-jogador1)))

```

A função `inverter-sinal-no` recebe como entrada um nó e utiliza uma função fornecida (`fn-inverter-sinal-jogo`) para inverter o tabuleiro, ou seja, trocar as posições dos jogadores -1 e -2. Ao criar um novo nó com o tabuleiro invertido, ela realiza a troca dos pontos dos jogadores, mantendo inalterado o valor de utilidade.

Esta função é particularmente útil quando o nó representa um jogador MIN, exigindo a inversão do tabuleiro. Como o algoritmo sempre joga com a mesma peça, é necessário alternar as posições dos jogadores para que o nó MIN seja jogado com o jogador 2.

```

(defun inverter-sinal-no (no fn-inverter-sinal-jogo)
  (let* ((novos-pontos-j1 (no-jogador2 no))
    (novos-pontos-j2 (no-jogador1 no)))

    (criar-no (funcall fn-inverter-sinal-jogo (no-tabuleiro no))
      (no-pai no)
      (no-f no)
      novos-pontos-j1
      novos-pontos-j2)))

```

Análise dos Resultados e Limitações

As principais limitações do algoritmo residem na ausência da implementação de memoização, isto é, a utilização de uma tabela de *hash* para armazenar os valores dos nós. Esta abordagem permitiria que, ao encontrar um nó já analisado anteriormente, o algoritmo evitasse explorar a árvore até as folhas para obter o valor do nó, realizando uma simples pesquisa nos valores armazenados na tabela *hash*.

Outra lacuna importante é a falta de implementação da busca quiescente, que poderia contribuir significativamente para a identificação da melhor jogada. Atualmente, o algoritmo tende a escolher a casa com a pontuação mais alta, uma estratégia válida para maximizar a pontuação. No entanto, essa abordagem pode falhar se a jogada resultar em poucas opções ou se essas opções tiverem pontuações baixas, comprometendo a eficácia do algoritmo.

A não implementação desses mecanismos decorre da gestão desafiadora do tempo disponível, levando-nos a concentrar apenas na implementação do algoritmo principal.

Durante a execução do programa, ao escolher a opção de jogo entre computadores, notamos que, à medida que as pontuações aumentam, ocorrem alguns erros de cálculo na pontuação.

Análise Estatística

De seguida, é apresentada uma análise estatística de uma partida *Humano vs Computador* recorrendo aos dados arquivados no ficheiro *log.dat* durante esse jogo.

Neste *log*, aquando da pontuação, J1 refere-se ao Computador (CPU) enquanto que J2 refere-se ao Humano (Utilizador). As peças do tabuleiro -1 e -2 correspondem, respetivamente, ao CPU e ao Utilizador.

O tempo limite atribuído à jogada do CPU foi de 1000 milissegundos (1 segundo).

```
o ----- o

- Jogo do Cavalo -
Partida iniciada.

Modo de Jogo: Humano vs CPU
1.o a jogar: HUMANO
Tempo limite do CPU: 1000 ms

o ----- o

-----

      0    1    2    3    4    5    6    7    8    9

0    90    79    68    40    -1    54    35    0    45    48
1    81    44    53    15    71    69    25    56    64    99
2    23    58    14    13    61    18    84    63    34    96
3    28    9     51    29    7     12    24    4     73    NIL
4    42    32    38    NIL    16    97    1     83    37    27
5    76    47    5     30    8     33    3     65    74    89
6    87    60    17    36    52    10    72    88    43    66
7    94    67    -2    85    98    49    75    95    21    22
8    46    77    41    55    57    92    2     70    80    82
9    78    NIL    31    11    50    86    59    6     20    NIL

O Jogador -2 jogou na posicao (7, 2).
Pontos atuais: J1 - 91 pontos | J2 - 155 pontos
Nos analisados: NIL
Numero de cortes: NIL
```

Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	90	79	68	40	NIL	54	NIL	0	45	48
1	81	44	-1	15	71	69	25	56	64	99
2	23	58	14	13	61	18	84	63	34	96
3	28	9	51	29	7	12	24	4	73	NIL
4	42	32	38	NIL	16	97	1	83	37	27
5	76	47	5	30	8	33	3	65	74	89
6	87	60	17	36	52	10	72	88	43	66
7	94	67	-2	85	98	49	75	95	21	22
8	46	77	41	55	57	92	2	70	80	82
9	78	NIL	31	11	50	86	59	6	20	NIL

O Jogador -1 jogou na posicao (1, 2).

Pontos atuais: J1 - 144 pontos | J2 - 155 pontos

Nos analisados: 162

Numero de cortes: 5

Duracao da jogada: 2

	0	1	2	3	4	5	6	7	8	9
0	90	79	68	40	NIL	54	NIL	0	45	48
1	81	44	-1	15	71	69	25	56	64	99
2	23	58	14	13	61	18	84	63	34	96
3	28	9	51	29	7	12	24	4	73	NIL
4	42	32	38	NIL	16	97	1	83	37	27
5	76	47	5	-2	8	33	NIL	65	74	89
6	87	60	17	36	52	10	72	88	43	66
7	94	67	NIL	85	98	49	75	95	21	22
8	46	77	41	55	57	92	2	70	80	82
9	78	NIL	31	11	50	86	59	6	20	NIL

O Jogador -2 jogou na posicao (5, 3).

Pontos atuais: J1 - 144 pontos | J2 - 185 pontos

Nos analisados: NIL

Numero de cortes: NIL

Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
--	---	---	---	---	---	---	---	---	---	---

0	90	79	68	40	NIL	54	NIL	0	45	48
1	81	44	NIL	15	71	69	25	56	64	99
2	23	58	14	13	61	18	84	63	34	96
3	28	9	51	-1	7	12	24	4	73	NIL
4	42	32	38	NIL	16	97	1	83	37	27
5	76	47	5	-2	8	33	NIL	65	74	89
6	87	60	17	36	52	10	72	88	43	66
7	94	67	NIL	85	98	49	75	95	21	22
8	46	77	41	55	57	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	6	20	NIL

O Jogador -1 jogou na posicao (3, 3).

Pontos atuais: J1 - 173 pontos | J2 - 185 pontos

Nos analisados: 21

Numero de cortes: 0

Duracao da jogada: 1

	0	1	2	3	4	5	6	7	8	9
0	90	79	68	40	NIL	54	NIL	0	45	48
1	81	44	NIL	15	71	69	25	56	64	99
2	23	58	14	13	61	18	84	63	34	96
3	28	9	51	-1	7	12	24	4	73	NIL
4	42	32	38	NIL	16	97	1	83	37	27
5	76	47	5	NIL	8	33	NIL	65	74	NIL
6	87	60	17	36	52	10	72	88	43	66
7	94	67	NIL	85	-2	49	75	95	21	22
8	46	77	41	55	57	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	6	20	NIL

O Jogador -2 jogou na posicao (7, 4).

Pontos atuais: J1 - 173 pontos | J2 - 283 pontos

Nos analisados: NIL

Numero de cortes: NIL

Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	90	NIL	68	40	NIL	54	NIL	0	45	48
1	81	44	NIL	15	71	69	25	56	64	99
2	23	58	14	13	61	18	84	63	34	96
3	28	9	51	NIL	7	12	24	4	73	NIL
4	42	32	38	NIL	16	-1	1	83	37	27
5	76	47	5	NIL	8	33	NIL	65	74	NIL
6	87	60	17	36	52	10	72	88	43	66
7	94	67	NIL	85	-2	49	75	95	21	22

8	46	77	41	55	57	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	6	20	NIL

O Jogador -1 jogou na posicao (4, 5).

Pontos atuais: J1 - 270 pontos | J2 - 283 pontos

Nos analisados: 114

Numero de cortes: 6

Duracao da jogada: 2

	0	1	2	3	4	5	6	7	8	9
0	90	NIL	68	40	NIL	54	NIL	0	45	48
1	81	44	NIL	15	71	69	25	56	64	99
2	23	58	14	13	61	18	84	63	34	96
3	28	9	51	NIL	7	12	24	4	73	NIL
4	42	32	38	NIL	16	-1	1	83	37	NIL
5	76	47	5	NIL	8	33	NIL	65	74	NIL
6	87	60	17	36	52	10	-2	88	43	66
7	94	67	NIL	85	NIL	49	75	95	21	22
8	46	77	41	55	57	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	6	20	NIL

O Jogador -2 jogou na posicao (6, 6).

Pontos atuais: J1 - 270 pontos | J2 - 355 pontos

Nos analisados: NIL

Numero de cortes: NIL

Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	90	NIL	68	40	NIL	54	NIL	0	45	NIL
1	81	44	NIL	15	71	69	25	56	64	99
2	23	58	14	13	61	18	-1	63	34	96
3	28	9	51	NIL	7	12	24	4	73	NIL
4	42	32	38	NIL	16	NIL	1	83	37	NIL
5	76	47	5	NIL	8	33	NIL	65	74	NIL
6	87	60	17	36	52	10	-2	88	43	66
7	94	67	NIL	85	NIL	49	75	95	21	22
8	46	77	41	55	57	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	6	20	NIL

O Jogador -1 jogou na posicao (2, 6).

Pontos atuais: J1 - 354 pontos | J2 - 355 pontos

Nos analisados: 82

Numero de cortes: 0

Duracao da jogada: 1

	0	1	2	3	4	5	6	7	8	9
0	90	NIL	68	40	NIL	54	NIL	0	45	NIL
1	81	44	NIL	15	71	69	25	56	64	99
2	23	58	14	13	61	18	-1	63	34	96
3	28	9	51	NIL	7	NIL	24	4	73	NIL
4	42	32	38	NIL	16	NIL	1	83	37	NIL
5	76	47	5	NIL	8	33	NIL	65	74	NIL
6	87	60	17	36	52	10	NIL	88	43	66
7	94	67	NIL	85	NIL	49	75	95	-2	22
8	46	77	41	55	57	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	6	20	NIL

O Jogador -2 jogou na posicao (7, 8).

Pontos atuais: J1 - 354 pontos | J2 - 376 pontos

Nos analisados: NIL

Numero de cortes: NIL

Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	90	NIL	68	40	NIL	54	NIL	0	45	NIL
1	81	44	NIL	15	71	69	25	56	64	99
2	23	58	14	13	61	18	NIL	63	34	96
3	28	9	51	NIL	-1	NIL	24	4	73	NIL
4	42	32	38	NIL	16	NIL	1	83	37	NIL
5	76	47	5	NIL	8	33	NIL	65	74	NIL
6	87	60	17	36	52	10	NIL	88	43	66
7	94	67	NIL	85	NIL	49	75	95	-2	22
8	46	77	41	55	57	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	6	20	NIL

O Jogador -1 jogou na posicao (3, 4).

Pontos atuais: J1 - 361 pontos | J2 - 376 pontos

Nos analisados: 102

Numero de cortes: 4

Duracao da jogada: 1

	0	1	2	3	4	5	6	7	8	9
0	90	NIL	68	40	NIL	54	NIL	0	45	NIL

1	81	44	NIL	15	71	69	25	56	64	99
2	23	58	14	13	61	18	NIL	63	34	96
3	28	9	51	NIL	-1	NIL	24	4	73	NIL
4	42	32	38	NIL	16	NIL	1	83	37	NIL
5	76	47	5	NIL	8	33	NIL	65	74	NIL
6	87	60	17	36	52	10	NIL	88	43	66
7	94	67	NIL	85	NIL	49	75	95	NIL	22
8	46	77	41	55	57	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	-2	20	NIL

O Jogador -2 jogou na posicao (9, 7).

Pontos atuais: J1 - 361 pontos | J2 - 382 pontos

Nos analisados: NIL

Numero de cortes: NIL

Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	90	NIL	68	40	NIL	54	NIL	0	45	NIL
1	81	44	NIL	-1	71	69	25	56	64	99
2	23	58	14	13	61	18	NIL	63	34	96
3	28	9	NIL	NIL	NIL	NIL	24	4	73	NIL
4	42	32	38	NIL	16	NIL	1	83	37	NIL
5	76	47	5	NIL	8	33	NIL	65	74	NIL
6	87	60	17	36	52	10	NIL	88	43	66
7	94	67	NIL	85	NIL	49	75	95	NIL	22
8	46	77	41	55	57	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	-2	20	NIL

O Jogador -1 jogou na posicao (1, 3).

Pontos atuais: J1 - 376 pontos | J2 - 382 pontos

Nos analisados: 95

Numero de cortes: 0

Duracao da jogada: 1

	0	1	2	3	4	5	6	7	8	9
0	90	NIL	68	40	NIL	54	NIL	0	45	NIL
1	81	44	NIL	-1	71	69	25	56	64	99
2	23	58	14	13	61	18	NIL	63	34	96
3	28	9	NIL	NIL	NIL	NIL	24	4	73	NIL
4	42	32	38	NIL	16	NIL	1	83	37	NIL
5	76	47	5	NIL	8	33	NIL	65	74	NIL
6	87	60	17	36	52	10	NIL	88	43	66
7	94	67	NIL	85	NIL	49	-2	95	NIL	22
8	46	77	41	55	NIL	NIL	2	70	80	82

9 78 NIL 31 11 50 86 59 NIL 20 NIL

O Jogador -2 jogou na posicao (7, 6).

Pontos atuais: J1 - 376 pontos | J2 - 457 pontos

Nos analisados: NIL

Numero de cortes: NIL

Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	90	NIL	68	40	NIL	54	NIL	0	45	NIL
1	81	44	NIL	NIL	71	69	25	56	64	99
2	23	-1	14	13	61	18	NIL	63	34	96
3	28	9	NIL	NIL	NIL	NIL	24	4	73	NIL
4	42	32	38	NIL	16	NIL	1	83	37	NIL
5	76	47	5	NIL	8	33	NIL	65	74	NIL
6	87	60	17	36	52	10	NIL	88	43	66
7	94	67	NIL	NIL	NIL	49	-2	95	NIL	22
8	46	77	41	55	NIL	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	NIL	20	NIL

O Jogador -1 jogou na posicao (2, 1).

Pontos atuais: J1 - 434 pontos | J2 - 457 pontos

Nos analisados: 10

Numero de cortes: 0

Duracao da jogada: 1

	0	1	2	3	4	5	6	7	8	9
0	90	NIL	68	40	NIL	54	NIL	0	45	NIL
1	81	44	NIL	NIL	71	69	25	NIL	64	99
2	23	-1	14	13	61	18	NIL	63	34	96
3	28	9	NIL	NIL	NIL	NIL	24	4	73	NIL
4	42	32	38	NIL	16	NIL	1	83	37	NIL
5	76	47	5	NIL	8	33	NIL	-2	74	NIL
6	87	60	17	36	52	10	NIL	88	43	66
7	94	67	NIL	NIL	NIL	49	NIL	95	NIL	22
8	46	77	41	55	NIL	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	NIL	20	NIL

O Jogador -2 jogou na posicao (5, 7).

Pontos atuais: J1 - 434 pontos | J2 - 522 pontos

Nos analisados: NIL

Numero de cortes: NIL

Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	-1	NIL	68	40	NIL	54	NIL	0	45	NIL
1	81	44	NIL	NIL	71	69	25	NIL	64	99
2	23	NIL	14	13	61	18	NIL	63	34	96
3	28	NIL	NIL	NIL	NIL	NIL	24	4	73	NIL
4	42	32	38	NIL	16	NIL	1	83	37	NIL
5	76	47	5	NIL	8	33	NIL	-2	74	NIL
6	87	60	17	36	52	10	NIL	88	43	66
7	94	67	NIL	NIL	NIL	49	NIL	95	NIL	22
8	46	77	41	55	NIL	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	NIL	20	NIL

O Jogador -1 jogou na posicao (0, 0).
Pontos atuais: J1 - 524 pontos | J2 - 522 pontos
Nos analisados: 9
Numero de cortes: 0
Duracao da jogada: 1

	0	1	2	3	4	5	6	7	8	9
0	-1	NIL	68	40	NIL	54	NIL	0	45	NIL
1	81	44	NIL	NIL	71	69	25	NIL	64	99
2	23	NIL	14	13	61	18	NIL	63	34	96
3	28	NIL	NIL	NIL	NIL	NIL	24	4	-2	NIL
4	42	32	38	NIL	16	NIL	1	83	NIL	NIL
5	76	47	5	NIL	8	33	NIL	NIL	74	NIL
6	87	60	17	36	52	10	NIL	88	43	66
7	94	67	NIL	NIL	NIL	49	NIL	95	NIL	22
8	46	77	41	55	NIL	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	NIL	20	NIL

O Jogador -2 jogou na posicao (3, 8).
Pontos atuais: J1 - 524 pontos | J2 - 595 pontos
Nos analisados: NIL
Numero de cortes: NIL
Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	-1	NIL	68	40	NIL	54	NIL	0	45	NIL
1	81	NIL	NIL	NIL	71	69	25	NIL	64	-2

2	23	NIL	14	13	61	18	NIL	63	34	96
3	28	NIL	NIL	NIL	NIL	NIL	24	4	NIL	NIL
4	42	32	38	NIL	16	NIL	1	83	NIL	NIL
5	76	47	5	NIL	8	33	NIL	NIL	74	NIL
6	87	60	17	36	52	10	NIL	88	43	66
7	94	67	NIL	NIL	NIL	49	NIL	95	NIL	22
8	46	77	41	55	NIL	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	NIL	20	NIL

O Jogador -2 jogou na posicao (1, 9).

Pontos atuais: J1 - 524 pontos | J2 - 694 pontos

Nos analisados: NIL

Numero de cortes: NIL

Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	-1	NIL	68	40	NIL	54	NIL	0	45	NIL
1	81	NIL	NIL	NIL	71	69	25	NIL	64	NIL
2	23	NIL	14	13	61	18	NIL	-2	34	96
3	28	NIL	NIL	NIL	NIL	NIL	24	4	NIL	NIL
4	42	32	38	NIL	16	NIL	1	83	NIL	NIL
5	76	47	5	NIL	8	33	NIL	NIL	74	NIL
6	87	60	17	NIL	52	10	NIL	88	43	66
7	94	67	NIL	NIL	NIL	49	NIL	95	NIL	22
8	46	77	41	55	NIL	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	NIL	20	NIL

O Jogador -2 jogou na posicao (2, 7).

Pontos atuais: J1 - 524 pontos | J2 - 757 pontos

Nos analisados: NIL

Numero de cortes: NIL

Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	-1	NIL	68	40	NIL	NIL	NIL	0	-2	NIL
1	81	NIL	NIL	NIL	71	69	25	NIL	64	NIL
2	23	NIL	14	13	61	18	NIL	NIL	34	96
3	28	NIL	NIL	NIL	NIL	NIL	24	4	NIL	NIL
4	42	32	38	NIL	16	NIL	1	83	NIL	NIL
5	76	47	5	NIL	8	33	NIL	NIL	74	NIL
6	87	60	17	NIL	52	10	NIL	88	43	66
7	94	67	NIL	NIL	NIL	49	NIL	95	NIL	22
8	46	77	41	55	NIL	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	NIL	20	NIL

O Jogador -2 jogou na posicao (0, 8).
Pontos atuais: J1 - 524 pontos | J2 - 802 pontos
Nos analisados: NIL
Numero de cortes: NIL
Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	-1	NIL	68	40	NIL	NIL	NIL	0	NIL	NIL
1	81	NIL	NIL	NIL	71	NIL	25	NIL	64	NIL
2	23	NIL	14	13	61	18	NIL	NIL	34	-2
3	28	NIL	NIL	NIL	NIL	NIL	24	4	NIL	NIL
4	42	32	38	NIL	16	NIL	1	83	NIL	NIL
5	76	47	5	NIL	8	33	NIL	NIL	74	NIL
6	87	60	17	NIL	52	10	NIL	88	43	66
7	94	67	NIL	NIL	NIL	49	NIL	95	NIL	22
8	46	77	41	55	NIL	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	NIL	20	NIL

O Jogador -2 jogou na posicao (2, 9).
Pontos atuais: J1 - 524 pontos | J2 - 898 pontos
Nos analisados: NIL
Numero de cortes: NIL
Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	-1	NIL	68	40	NIL	NIL	NIL	0	NIL	NIL
1	81	NIL	NIL	NIL	71	NIL	25	NIL	64	NIL
2	23	NIL	14	13	61	18	NIL	NIL	34	NIL
3	28	NIL	NIL	NIL	NIL	NIL	24	-2	NIL	NIL
4	42	32	38	NIL	16	NIL	1	83	NIL	NIL
5	76	47	5	NIL	8	33	NIL	NIL	74	NIL
6	87	60	17	NIL	52	10	NIL	88	43	66
7	94	67	NIL	NIL	NIL	49	NIL	95	NIL	22
8	46	77	41	55	NIL	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	NIL	20	NIL

O Jogador -2 jogou na posicao (3, 7).
Pontos atuais: J1 - 524 pontos | J2 - 902 pontos
Nos analisados: NIL
Numero de cortes: NIL
Duracao da jogada: NIL

	0	1	2	3	4	5	6	7	8	9
0	-1	NIL	68	40	NIL	NIL	NIL	0	NIL	NIL
1	81	NIL	NIL	NIL	71	NIL	25	NIL	-2	NIL
2	23	NIL	14	13	61	18	NIL	NIL	34	NIL
3	28	NIL	NIL	NIL	NIL	NIL	24	NIL	NIL	NIL
4	42	32	38	NIL	16	NIL	1	83	NIL	NIL
5	76	47	5	NIL	8	33	NIL	NIL	74	NIL
6	87	60	17	NIL	52	10	NIL	88	43	66
7	94	67	NIL	NIL	NIL	49	NIL	95	NIL	22
8	NIL	77	41	55	NIL	NIL	2	70	80	82
9	78	NIL	31	11	50	86	59	NIL	20	NIL

O Jogador -2 jogou na posicao (1, 8).

Pontos atuais: J1 - 524 pontos | J2 - 966 pontos

Nos analisados: NIL

Numero de cortes: NIL

Duracao da jogada: NIL

O ----- O

- Jogo do Cavalo -
Partida terminada.

O vencedor e: Utilizador!

CPU: 524 pontos
Utilizador: 966 pontos

O ----- O